AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) Method of making secure the execution of a computer program (EXE) including a set of instructions comprising at least one instruction, which method is characterized in that it includes:

- a first step (E30), prior to the execution of the computer program, of calculating and storing a first signature (SIG1) representative of the intended execution of the set of instructions,

- a second step (E50), during the execution of the set of instructions, of calculating and storing a second signature (SIG2) representative of the execution of the set of instructions, and

- a step (E60) of detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2),

wherein said set of instructions comprising at least one critical instruction in the form of a jump instruction of any type (JMP, JNZ, CJNE, JZ) within the sequence of instructions of said set of instructions.

2. (original) Method according to claim 1, characterized in that the first calculation and storage step (E30) is executed during the generation of the instructions (A1, A13) of the computer program.

3. (previously presented) Method according to claim 1, characterized in that the second signature (SIG2) stored during the second calculation and storage step (E50) is retained in memory during the execution of at least one second instruction following the set of instructions.

4. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) is obtained from the number of instructions in the set of instructions,

- the second signature (SIG2) is obtained from the number of instructions from the set of instructions that have been executed, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

5. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) is obtained from the number of instructions in the set of instructions,

- the second signature (SIG2) is obtained from the number of instructions from the set of instructions that have not been executed, this second signature (SIG2) being calculated from the first signature (SIG1), and in that

- the detection step (E60) detects an execution anomaly when the value of the second signature (SIG2) is not zero after the execution of the set of instructions.

6. (original) Method according to claim 5, characterized in that an interrupt of the computer program is triggered when the value of the second signature (SIG2) is below a predetermined threshold.

7. (previously presented) Method according to claim 5, characterized in that the first signature (SIG1) and the second signature (SIG2) are retained in memory during the execution of the program in the same register (REG1).

8. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) is obtained from the code of a critical instruction of the set of instructions,

- the second signature is obtained from the code of the critical instruction, that code being stored at the same time as or after the execution of the critical instruction, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

9. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) is obtained from the address of a critical instruction of the set of instructions, the address being obtained during or after the generation of the executable code of the set of instructions,

- the second signature (SIG2) is obtained from the address of the critical instruction, that address being stored (E30) at the same time as or after the execution (E30) of the critical instruction, and

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

10. (cancelled)

11. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) and the second signature (SIG2) are error detector codes (CRC1, CRC2) calculated from the code or from an address of an instruction of the set of instructions, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

12. (original) Method according to claim 11, characterized in that the error detector codes (CRC1, CRC2) are cyclic redundancy check codes.

13. (original) Method according to claim 11, characterized in that the error detector codes are obtained by the logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

14. (previously presented) Method according to claim 1, characterized in that:

- the first signature (SIG1) and the second signature (SIG2) are respectively obtained during the generation and the execution of the instructions from at least two elements chosen from:

. the number of instructions in the set of instructions,

. the code of at least one instruction of the set of instructions,

. the address of at least one instruction of the set of instructions, and

. an error detector code calculated from the code or an address of at least one critical instruction of the set of instructions, the address being obtained during or after the generation of the executable code of the set of instructions, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

15. (previously presented) Method according to claim 1, characterized in that it includes a step (E70) of destroying at least a portion of the system on which the computer program is executed, this step of destroying being made when an execution anomaly is detected in the detection step.

16. (previously presented) Method according to claim 1, characterized in that the first signature (SIG1) is generated automatically (E30).

17. (currently amended) Device for processing a computer program including a set of at least one instruction, characterized in that it includes means (12) for calculating and storing ~~the~~ a first signature (SIG1), the first signature (SIG1) stored in a memory and the first signature (SIG1) is representative of the intended execution of the set of instructions prior to the execution thereof, said set of instructions comprising at least one critical instruction in the form of a jump instruction of any type (JMP, JNZ, CJNE, JZ) within the sequence of instructions of said set of instructions.

18. (original) Device according to claim 17, characterized in that the means (12) for calculating and storing the first signature (SIG1) are adapted to calculate and store information obtained from the number of instructions of the set of instructions.

19. (original) Device according to claim 17, characterized in that the means (12) for calculating and storing the first signature (SIG1) are adapted to obtain and store information obtained from the code of a critical instruction of the set of instructions.

20. (previously presented) Device according to claim 17, characterized in that it further includes means (14) for generating executable code from the computer program (SOURCE).

21. (original) Device according to claim 20, characterized in that the means for calculating and storing the first signature (SIG1) are adapted to obtain and store information obtained from the address of a critical instruction, the information being obtained of the set of instructions by the means (14) for generating executable code.

22. (cancelled)

23. (original) Device according to claim 17, characterized in that the means (12) for calculating and storing the first signature (SIG1) are adapted to calculate and store information obtained from an error detector code (CRC1) calculated from the code or an address of at least one instruction of the set of instructions.

24. (original) Device according to claim 23, characterized in that the error detector code (CRC1) is a cyclic redundancy check code.

25. (original) Device according to claim 23, characterized in that the error detector code is obtained by a logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

26. (currently amended) Device for making secure the execution of a computer program including a set of instructions comprising at least one instruction, which device is characterized in that it includes:

- a first register (REG1) for storing a first signature (SIG1) representative of the intended execution of the set of instructions,

- means (22) for calculating and storing in said first register (REG1) or in a second storage register (REG2) during the execution of the set of instructions a second signature (SIG2) representative of the execution of the set of instructions, and

- means (24) for detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2),

said set of instructions comprising at least one critical instruction in the form of a jump instruction of any type (JMP, JNZ, CJNE, JZ) within the sequence of instructions of said set of instructions.

27. (original) Device according to claim 26, characterized in that the calculation and storage means are adapted to retain the second signature (SIG2) in the second register (REG2) during the execution of at least one second instruction following the set of instructions.

28. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) being obtained from the number of instructions of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the number of instructions of the set of instructions that have been executed and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

29. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) being obtained from the number of instructions of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the number of instructions of the set of instructions that have not been executed, this second signature (SIG2) being calculated from the first signature (SIG1), and the in that detection means (24) detect an execution anomaly when the value of second signature (SIG2) is not zero after the execution of the set of instructions.

30. (original) Device according to claim 29, characterized in that it further includes means for triggering an interrupt of the computer program when the value of the second signature (SIG2) is below a predetermined threshold.

31. (cancelled)

32. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) being obtained from the code of a critical instruction of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the code of the critical instruction, the code being stored at the same time as or after the execution of the critical instruction, and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

33. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) being obtained from the address of a critical instruction of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the address of the critical instruction, that address being stored at the same time as or after the execution of the critical instruction, and in that the detection means detect an execution anomaly when the first and second signatures are different after the execution of the set of instructions.

34. (cancelled)

35. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) and the second signature (SIG2) being error detector codes (CRC1, CRC2) calculated from the code of an instruction of the set of instructions, the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

36. (original) Device according to claim 35, characterized in that the error detector codes (CRC1, CRC2) are cyclic redundancy check codes.

37. (original) Device according to claim 35, characterized in that the error detector codes are obtained by a logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

38. (previously presented) Device according to claim 26, characterized in that, the first signature (SIG1) being obtained from at least two elements chosen from:

- the number of instructions of the set of instructions,

- the code of at least one instruction of the set of instructions,

- the address of at least one instruction of the set of instructions, and

- an error detector code calculated from the code or the address of at least one instruction of the set of instructions,

the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained in a similar manner from the at least two elements during the execution of the instructions and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

39. (previously presented) Device according to claim 26, characterized in that it further includes means for destroying at least a portion of the computer program.

40. (previously presented) Microcircuit card characterized in that it includes a securing device (100) according to claim 26.